Comput. Chem. Vol. 12, No. 4, pp. 343–356, 1988
Printed in Great Britain. All rights reserved

0097-8485/88 $3.00 + 0.00
Copyright © 1988 Pergamon Press plc

APPLICATION

A GENERALIZATION FOR MIXTURES OF A FAST ALGORITHM TO CALCULATE SOME INTERMOLECULAR ORIENTATIONAL AVERAGES

S. LAGO* and C. VEGA

Departamento de Química Física, Facultad de Ciencias Químicas, Universidad Complutense, 28040 Madrid, Spain

(*Received* 19 *November* 1987)

Abstract—A previously reported algorithm has been generalized in order to calculate the shortest distances and orientational averages for systems composed of molecules, that can be modelled by rods of differing lengths assumed to interact via an intermolecular potential that depends solely upon the shortest distance between the rods. The orientational averages calculated for each rod–rod interaction are the Boltzmann factor and two factors related to contributions to internal energy and Helmholtz free energy. All three factors have appeared in recent theories dealing with mixtures of polyatomic fluids, and the method presented here seems to open broader fields of applications for such theories.

1. INTRODUCTION

In a previous paper (Sevilla & Lago, 1985), hereinafter referred to as I, we have presented a fast algorithm to calculate the shortest distances and the orientational averages for rods. The rods model can be used to represent the behavior of linear molecules. In paper I, we postulated that our algorithm should enable us to undertake the study of pure liquids composed of rod-like molecules. These hopes have already been fulfilled in part, and initial results have been obtained using integral equations (Lago *et al.*, 1986; Lago & Sevilla, 1988) and by perturbation theories (Lago *et al.*, 1985; Lago & Padilla, 1987). Generalization of our method allows us to treat nonlinear molecules that are effectively rigid. These can be treated as sets of rods of uniform size. Examples are ozone, propane, cyclopropane, etc. Furthermore, a similar generalization has also been validated by Fischer & Lustig (1985) and Lustig (1986) who obtained good equations of state for a reference system for propane and cyclopropane.

A more complicated case is to model systems made up of segments of differing length. This interesting case can represent polyatomic molecules and mixtures of rod-like molecules. The first step in extending the algorithm is to consider mixtures as being composed either of monatomic + linear molecules or of linear molecules having different lengths. In this paper, we show how to extend our algorithm to such systems by a generalized algorithm for calculating shortest distances. This new algorithm is not only very convenient but also it is more efficient than our previous algorithm. The presentation is as follows: Section 2 describes the solution of the set of linear equations derived from our method and presents the definitions of physically meaningful quantities. Section 3 describes the program steps and the key variables. Section 4 presents representative results. A program listing is included in the Appendix.

2. COMPUTATION OF THE SHORTEST DISTANCE AND RELATED FUNCTIONS

Let us consider a system composed of two segments S_1 and S_2, as in our previous paper (Sevilla & Lago, 1985) except that now these segments may have different lengths L_1 and L_2. We choose a frame of reference coordinates whose origin is the geometric center (GC) of segment S_1 and whose z-axis is directed along this segment. Five coordinates are required to determine the position and the relative orientation of the second segment. These coordinates may, for example, be the polar coordinates of the GC of S_2, r, θ_1 and ϕ_1, and the coordinates of one end of S_2, θ_2 and ϕ_2 as shown in Fig. 1. Our first objective is to determine the shortest distance between these segments as a function of the Cartesian coordinates of the ends denoted as (a_i, b_i, c_i) and (e_i, f_i, g_i) where $i = 1, 2$. We also define:

$$\alpha_i = e_i - a_i \tag{1a}$$

$$\beta_i = f_i - b_i \tag{1b}$$

$$\gamma_i = g_i - c_i. \tag{1c}$$

Obviously, in this coordinate system, $\alpha_1 = \beta_1 = 0$ and $\gamma_1 = L$.

* To whom correspondence should be addressed.

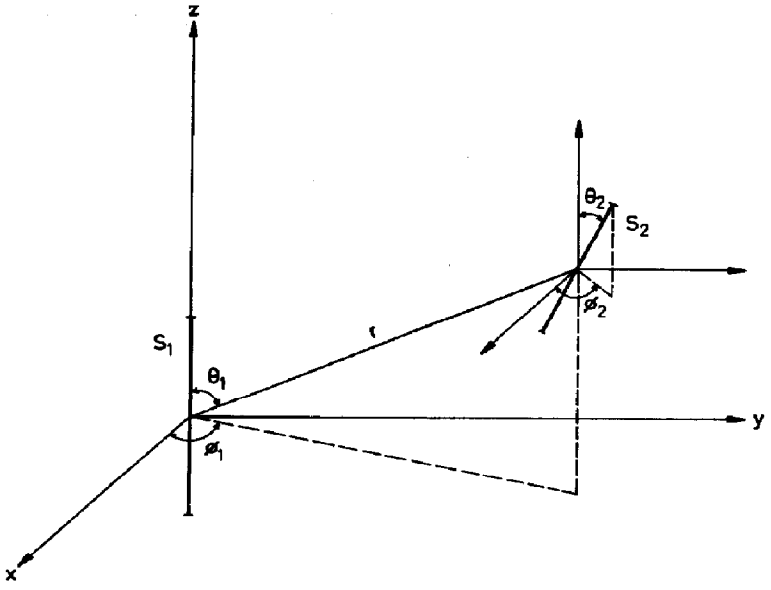

Fig. 1. Reference frames for the system of two rods of different lengths studied in this work.

The shortest distance, Δ, and the square of this distance are minima for the same pair of points. Therefore, we search for the minimum of the function:

$$\Delta^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 \qquad (2)$$

where (x_i, y_i, z_i) are points on S_i.

We will now proceed as in paper I, avoiding details and only noting here the essential steps of the method, namely:

(1) Division of segments (topollogically closed sets) in interiors (open sets) and ends (Lipschutz, 1965).

(2) Projection of the interior of each segment on the entire real axis (Lipschutz, 1965).

(3) Application of the Lagrange multiplier method to the projected sets.

(4) Solution of the coupled set of equations derived in Step 3 gives the shortest interior–interior distance (Δ_{ii}) and two parameters μ_1 and μ_2 which are indicators of whether or not the shortest distance is between interiors. If the shortest distance is between interiors ($0 < \mu_i < 1$) and $\Delta = \Delta_{ii}$, if not, the next steps are:

(5) The interior–end (Δ_{ie}) and end–end (Δ_{ee}) distances are calculated using standard geometric formulae.

(6) The lowest value of Δ_{ie} or Δ_{ee} is the shortest distance.

Solution of the set of equations in step 4 is:

$$x_1 = y_1 = 0 \qquad (3)$$

$$z_1 = z_2 \qquad (4)$$

$$x_2 = a_2 - [(\alpha_2\beta_2 b_2 + \alpha_2^2 a_2)/(L_2^2 - \gamma_2^2)] \qquad (5)$$

$$y_2 = b_2 - [(\alpha_2\beta_2 a_2 + \beta_2^2 b_2)/(L_2^2 - \gamma_2^2)] \qquad (6)$$

$$\mu_1 = (c_2 - c_1)/\gamma_1 - [(\beta_2 b_2\gamma_2 + a_2\alpha_2\gamma_2)/(L_2^2 - \gamma_2^2)]/\gamma_1 \qquad (7)$$

$$\mu_2 = -[(\alpha_2 a_2 + b_2\beta_2)/(L_2^2 - \gamma_2^2)] \qquad (8)$$

and the shortest distance between interiors is:

$$\Delta_{ii} = (x_2^2 + y_2^2)^{1/2}. \qquad (9)$$

We should point out, as in paper I, that if the two segments are parallel, $\gamma_2 = L_2$ and equation (8) is no longer valid. In this case, the shortest distance is either the distance between the straight lines containing the segments, when they share a common range of z-coordinates, or the distances between ends when this is not the case. The equations are standard in both these instances. In any other case, equations (3)–(9) are valid for mixtures of molecules modelled by segments of differing lengths (mixtures of linear molecules) and present two important special cases:

(a) If $L_1 = L_2$, equations (3)–(9) are reduced to those of paper I, namely, to a pure liquid modelled by only one type of segment.

(b) If $L_2 = 0$, the shortest distance corresponds to a distance between a point and a segment, and equations (3)–(9) are no longer valid. However, this distance like the distance between pairs of points is easily calculated using well-known geometric formulae. The system can, therefore, be a good reference model for mixtures of molecules with spherical symmetry and linear molecules; these mixtures can also be studied using standard formulae and equations (3)–(9).

Once the shortest distance has been calculated, the quantities required for a thermodynamical perturbation theory (Lago & Padilla, 1987) can be computed. These quantities are:

(1) The reference intermolecular potential:

$$u_{ij}^0(\Delta) = \begin{cases} 4\epsilon_{ij}[(\sigma_{ij}/\Delta_{ij})^{12} - (\sigma_{ij}/\Delta_{ij})^6] + \epsilon_{ij}, \\ \quad \text{if } \Delta_{ij} < 2^{1/6}\sigma_{ij} \qquad (10a) \\ 0, \quad \text{if } \Delta_{ij} > 2^{1/6}\sigma_{ij} \qquad (10b) \end{cases}$$

where the subscripts refer to the different intermolecular interactions.

(2) The Boltzmann factor of this potential:

$$e_{ij}^0 = \exp[-\beta u_{ij}^0(\Delta)] \qquad (11)$$

where $\beta = (kT)^{-1}$, k is the Boltzmann constant and T is the temperature in Kelvin.

(3) The product of the perturbation potential times the Boltzmann factor of the reference potential:

$$a_{ij} = u_{ij}^1(\Delta)\, e_{ij}^0(\Delta) \qquad (12)$$

where

$$u_{ij}^1(\Delta) = \begin{cases} -\epsilon_{ij}; \quad \Delta_{ij} < 2^{1/6}\sigma_{ij} \qquad (13a) \\ 4\epsilon_{ij}[(\sigma_{ij}/\Delta_{ij})^{12} - (\sigma_{ij}/\Delta_{ij})^6] \\ \quad \Delta_{ij} > 2^{1/6}\sigma_{ij}. \qquad (13b) \end{cases}$$

(4) The product of the complete interaction potential times the Boltzmann factor of the same potential:

$$v_{ij}(\Delta) = u_{ij}(\Delta) \exp[-\beta u_{ij}(\Delta)] \qquad (14)$$

where

$$u_{ij} = u_{ij}^0 + u_{ij}^1. \qquad (15)$$

Finally, we calculated the orientational averages for these quantities for constant distances between geometric centers for which:

$$\Delta = \Delta(r, \theta_1, \phi_1, \theta_2, \phi_2) \qquad (16)$$

with the averages defined as:

$$\langle x_{ij}(r)\rangle = \left\{ \int_0^{2\pi}\int_{-1}^{1}\int_0^{2\pi}\int_{-1}^{1} \times x_{ij}[\Delta(r, \theta_1, \phi_1, \theta_2, \phi_2)] \times \mathrm{d}[\cos(\theta_1)\,\mathrm{d}\phi_1 \cdot \mathrm{d}\cos(\theta_2)\,\mathrm{d}\phi_2] \Big/ \left[\int_0^{2\pi}\int_{-1}^{1}\int_0^{2\pi}\int_{-1}^{1} \times \mathrm{d}\cos(\theta_1)\,\mathrm{d}\phi_1 \cdot \mathrm{d}\cos(\theta_2)\,\mathrm{d}\phi_2 \right] \right\} \qquad (17)$$

where $x = \Delta$, u, e^0, a and v, are approximated by computing the mean value of the integrand using random numbers uniformly generated between -1 and 1 for the variables $\cos\theta_1$ and $\cos\theta_2$, and uniformly between 0 and 2π for the variables ϕ_1 and ϕ_2.

3. PROGRAM STRUCTURE

The computer program, named CICLOS, was run on an IBM 370/75 computer at the Computer Center, Complutense University, Madrid. A listing of CICLOS appears in the Appendix of this paper. Required inputs are the lengths of the rods, PL1 and PL2, potential parameters for like interactions PSIG1, PSIG2, PEPS1, and PEPS2. Parameters for unlike interaction are calculated from Lorentz–Berthellot rules:

$$\epsilon_{12} = (\epsilon_{11} \cdot \epsilon_{22})^{1/2} \qquad (18)$$

$$\sigma_{12} = (\sigma_{11} + \sigma_{22})/2. \qquad (19)$$

Other required input parameters are: (a) the temperature, TEMP, for which the averages are calculated for the functions defined by equations (11), (12) and (14), (b) the number of orientations required for the average computation in equation (17), NUPUNT, and (c) the number of GC–GC distances for which the averages are searched, NUCGCG.

The basic program steps are:

(1) Generation of a randomly mutual orientation of the rods. S_1 is chosen as the z-axis, with its geometric center (GC) as the origin of coordinates. The GC of the second rod is set at a distance R; the cosine of angle (CGCO) and the angle (FI), determining the position of this second GC, are also randomly generated. The Cartesian coordinates of this second GC are calculated and denoted by X, Y, Z. If the length of any of the rods is 0, the program jumps to step 4.

(2) Translation of the frame of reference coordinates to the point X, Y, Z. The variables $\cos\theta_2$ and ϕ_2, named SEGCO and FIPR, are randomly generated. These coordinates give the relative position of an end of S_2 with respect to the second frame of reference. The Cartesian coordinates of this end in the second frame of reference coordinates are named XPR, YPR, and ZPR.

(3) Cartesian coordinates of the ends of S_2 in the fixed laboratory frame of reference coordinates are calculated as:

$$\mathrm{A2} = \mathrm{X} + \mathrm{XPR}$$

$$\mathrm{B2} = \mathrm{Y} + \mathrm{YPR}$$

$$\mathrm{C2} = \mathrm{Z} + \mathrm{ZPR}$$

$$\mathrm{D2} = \mathrm{X} - \mathrm{XPR}$$

$$\mathrm{E2} = \mathrm{Y} - \mathrm{YPR}$$

$$\mathrm{F2} = \mathrm{Z} - \mathrm{ZPR}$$

and the components of a vector along this segment are:

$$\mathrm{ALFA2} = \mathrm{E2} - \mathrm{A2}$$

$$\mathrm{BETA2} = \mathrm{F2} - \mathrm{B2}$$

$$\mathrm{GAMMA2} = \mathrm{G2} - \mathrm{C2}.$$

Program execution jumps to step 5.

(4) MINDIS is calculated in the DISMLO subroutine using standard distance formulae either be-

tween a point and a straight line or between two points. From here, the program jumps to step 10.

(5) The rods are checked to see if they are parallel. If they are not, the program jumps to step 6. If they are and have a common z-interval, MINDIS is calculated by standard formulae and the program jumps to step 10. Otherwise, it jumps to step 7.

(6) The values of μ_1 (MU1) and μ_2 (MU2) are calculated by equations (7) and (8). If $0 < \mu_i < 1$ for both values of μ_i, then the minimum distance is interior–interior, MINDIS is computed using equations (5), (6) and (9), and the next program step is 10. If any value is not in the range between 0 and 1, MINDIS is not an interior–interior distance and the next step is 7.

(7) The four end–end distance values are calculated and stored as the first four values in array H, dimensioned as H (8).

(8) The four end–interior distance values are calculated and denoted as H(J), where J = 5, 6, 7 and 8. If the perpendicular from any end to the straight line defined by the other segment intersects this segment, the corresponding end–interior distance is stored in array H. If not, an arbitrarily large value is assigned to the array element, in our case:

$$H(J) = 1 \cdot E + 33.$$

(9) The minimum value of the elements in array H is found and this value is MINDIS.

(10) Once MINDIS has been found, the quantities defined by equations (11), (12), (14) and (15) are calculated in the UYEXP subroutine; and in the main program the variables for these values are EXP, U1EXPO, EXPTO, and U, respectively.

(11) The program then returns to step 1 and repeats the steps 1–10 NUPUNT times. The sums of the quantities calculated in 10 are stored as:

$$\text{SUMMID} = \text{SUMMID} + \text{MINDIS}$$
$$\text{SUMU} = \text{SUMU} + \text{U}$$
$$\text{SUMEXP} = \text{SUMEXP} + \text{EXP}$$
$$\text{SUU1EO} = \text{SUU1EO} + \text{U1EXPO}$$
$$\text{SUEXTO} = \text{SUEXTO} + \text{EXPTO}$$

(12) The averages are calculated in the program statements:

```
PROMID(CICLO, I)
    = SUMMID/NUPUNT
PROU(CICLO, I)
    = SUMU/NUPUNT/KBOLT/TEMP
PROEXP(CICLO, I)
    = SUMEXP/NUPUNT
PROU1E(CICLO, I)
    = SUU1EO/NUPUNT/KBOLT/TEMP
PROETO(CICLO, I)
    = SUEXTO/NUPUNT/KBOLT/TEMP
```

Table 1. Intermolecular potential parameters for the molecules studied in this work

Molecule	Ar*	N_2†	O_2‡
σ (nm)	0.3405	0.3207	0.3243
ϵ/k (K)	119.8	117	130
L (nm)	0	0.0930	0.0444

* Parameters taken from McDonald (1972).
† Parameters taken from Koide & Kihara (1974).
‡ Parameters taken from Diaz Peña *et al.* (1973).

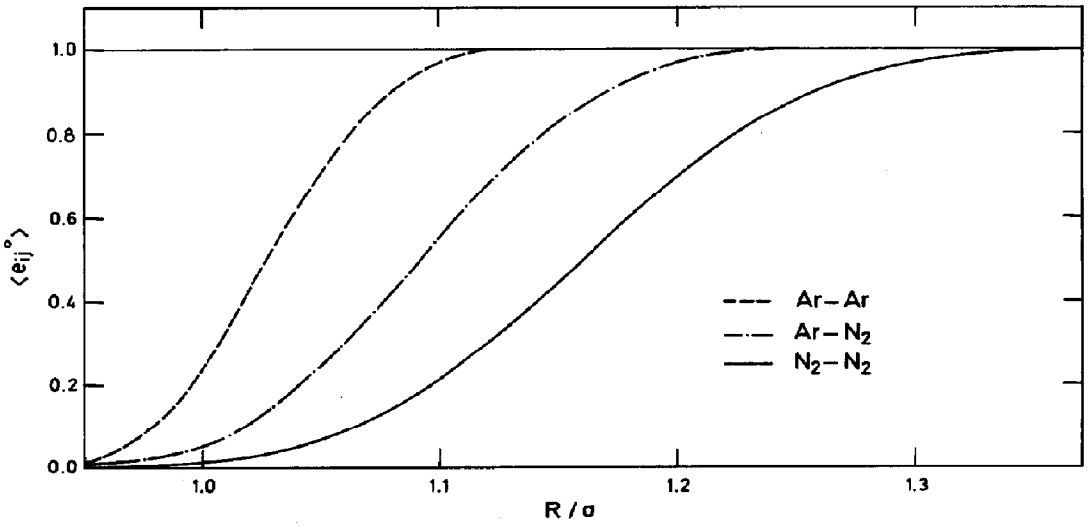

Fig. 2. Boltzmann factor at 84 K for the mixture $Ar + N_2$.

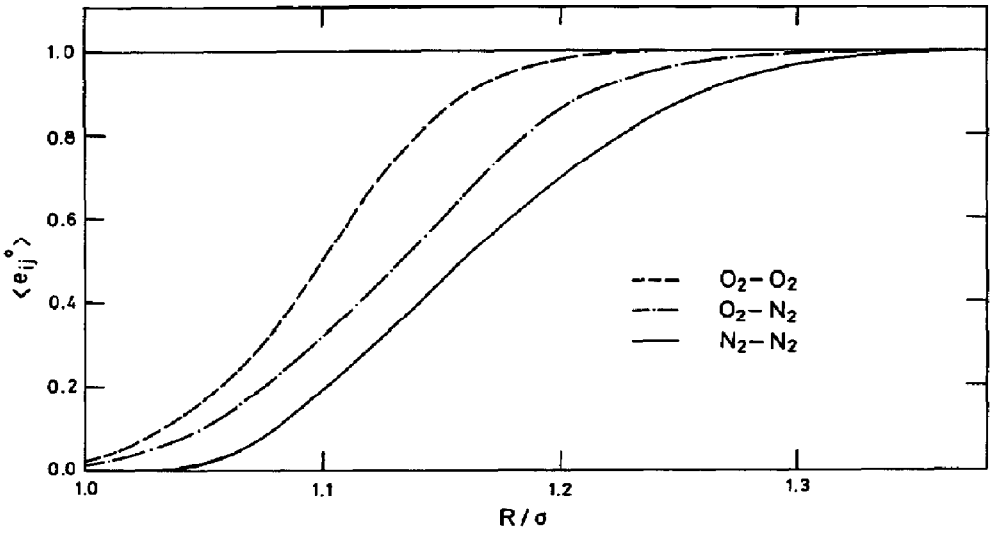

Fig. 3. As in Fig. 2 for the mixture $N_2 + O_2$.

where the index CICLO refers to the molecular interaction for which the averages are calculated and I refers to the GC–GC distance. KBOLT is the Boltzmann constant.

Lastly, the GC–GC distance is varied in the statement:

$$R = R + DELTAR * SIGX$$

and after NUGCGC iterations, the entire procedure is repeated for the next interaction.

4. RESULTS AND DISCUSSION

We normally employed 16,000 points to calculate the averages of the appropriate molecular interaction at each GC–GC distance. The computer used was slightly faster than the HP-1000 computer used for paper I. Moreover, our algorithm has now been made more efficient by eliminating the computation of unnecessary distances through the condition in step 6. As a result, the average for the rod–point inter-

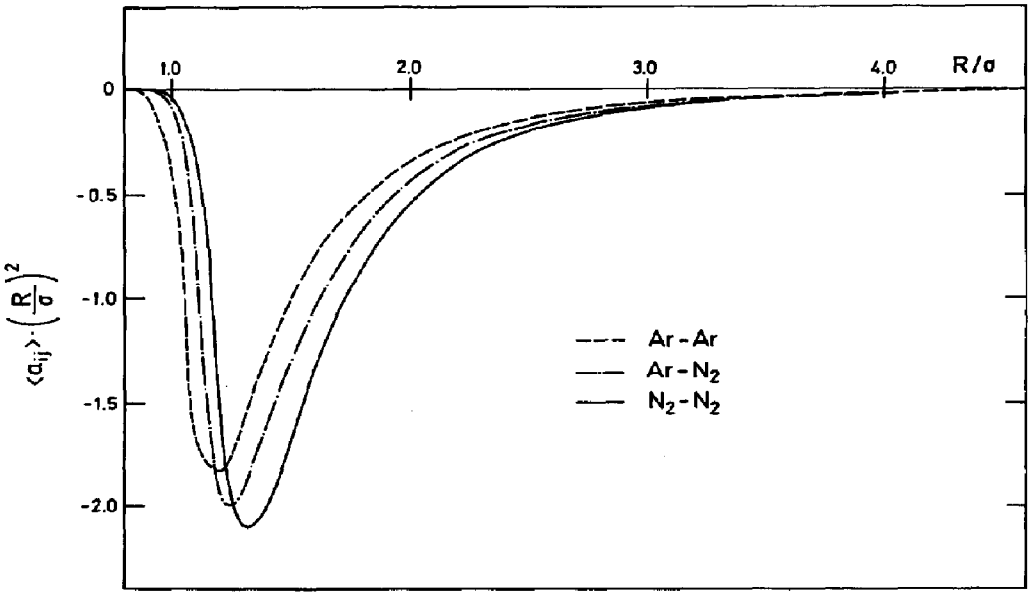

Fig. 4. Orientationally averaged values for the function defined in equation (12) of the text for the mixture $Ar + N_2$ at 84 K.

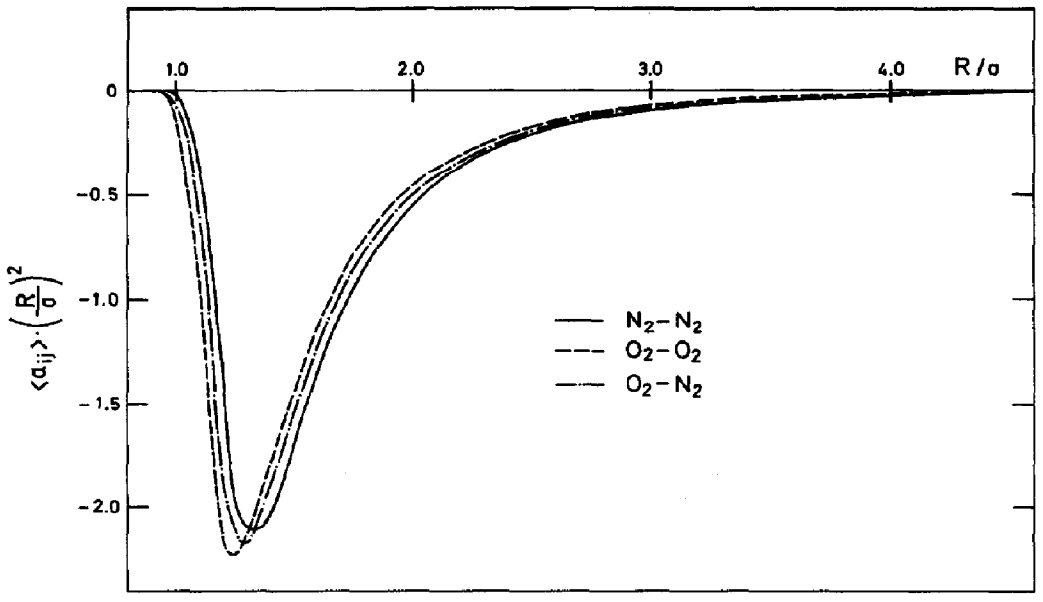

Fig. 5. As in Fig. 4 for the mixture $N_2 + O_2$.

action is calculated in about 1.5 min and the average for the rod–rod interaction is about 3 min. Accuracy is thought to be better than 1% and such levels of accuracy have been confirmed for selected runs using up to 50,000 points.

We applied our method of computation to a mixture of $Ar + N_2$ at 84 K and to a mixture $N_2 + O_2$ at the same temperature. The intermolecular potential parameters are given in Table 1. As far as we are aware, the previous computations reported for the first mixture have considered it to be either a mixture of monatomic molecules (Lee & Hulburt, 1973; Kohler, 1977) applying integral equations and perturbation theories, or a mixture of a 1-center-Lennard–Jones (ICLJ) monatomic molecule + a 2CLJ diatomic molecule applying different variants of perturbation theories (Enciso & Lombardero, 1981; Fischer & Lago, 1983). However, to our knowledge this is the first time computations have been reported for the potential given by equations (10), (13) and (15). For the second mixture, rigorous results obtained by perturbation theory have been reported (Bohn *et al.*, 1985) but also not for this potential.

Figures 2 and 3 depict the behavior of the Boltzmann factor in the mixtures studied. Despite the

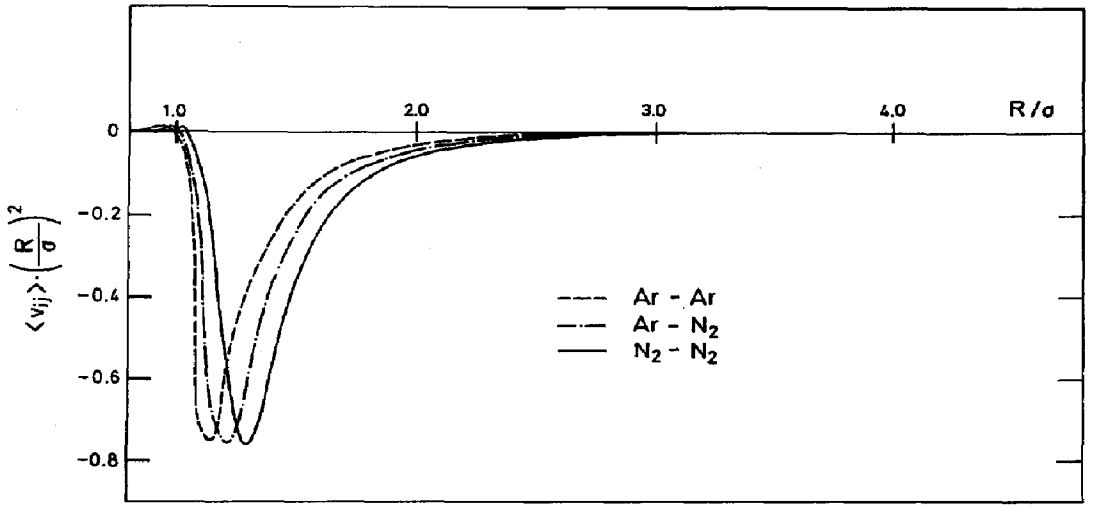

Fig. 6. As in Fig. 4, but for the function defined in equation (14).

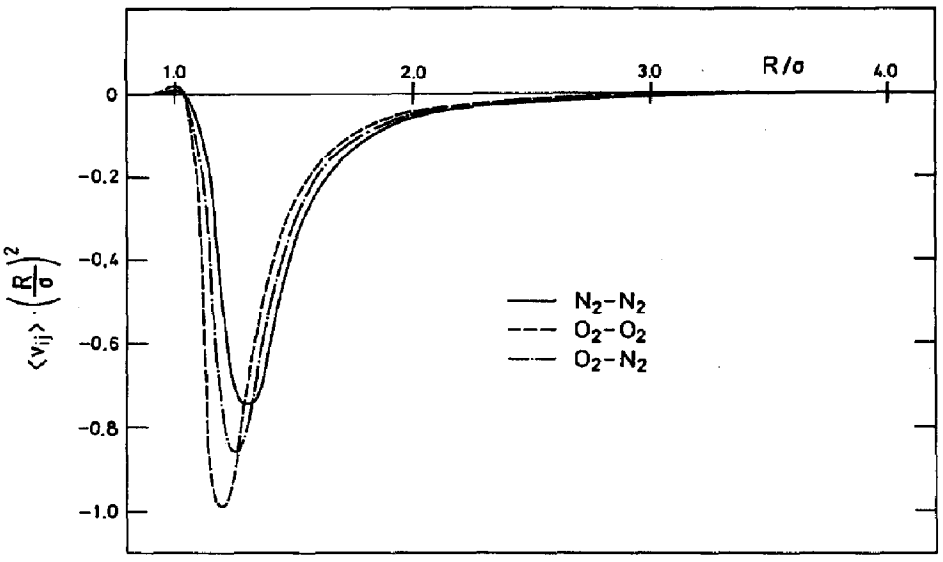

Fig. 7. As in Fig. 6 for the mixture $N_2 + O_2$.

small elongation of N_2 ($L^* = L/\sigma = 0.293$), the Boltzmann factor increases much more slowly for the N_2–N_2 interaction than for Ar–Ar interaction. Furthermore, the range where $\langle e^0_{12}(R) \rangle$ differs significantly from 0 or 1 is more than twice that for $\langle e^0_{11}(R) \rangle$. The relative success in modelling the mixture achieved in the past by a 1CLJ + 1CLJ system was very likely due to the chance cancellation of errors (Kohler, 1977). The more anisotropic the molecules, the more difficult such cancellation of errors becomes. Our method is not restricted by the molecular anisotropy, although the number of points needed to obtain accurate values for averaging increases with L^*. Differences in $\langle e^0_{ij}(R) \rangle$ are less apparent for $N_2 + O_2$, where the differences in L^* are also smaller.

Figures 4–7 show the variations in the averages defined by equations (12) and (14) for the two mixtures. These averages appear in perturbation theories in the calculation of the Helmholtz free energy and internal energy in some integrals where the integrand is a product of these averages times R^2 and times a pair correlation function (PCF). Because the value of the PCF is near 1, beyond 3 or 4σ, the contribution to these integrals ar large distances is given directly by the curves shown in the figures. It would seem clear that this contribution is negligible for values of $r > 5\sigma$ or lower, and the range in which the contribution is more important should be located between σ and 2σ where it is well-known that computational accuracy is more critical.

Acknowledgements—We wish to thank the help of Russell Sacks who considerably improved the English of our first version. This work was supported by Project PB0028/85 of the Spanish CAICYT (Comisión Asesora de Investigación Científica y Técnica).

REFERENCES

Bohn M., Lago S., Fischer J. & Kohler F. (1985) *Fluid Phase Equil.* **23**, 137.
Diaz Peña M., Renuncio J. A. R., Cabello A. & Sotomayor C. P. (1973) *An. R. Soc. Esp. Fis. Quim.* **69**, 163.
Enciso E. & Lombardero M. (1981) *Mol. Phys.* **44**, 725.
Fischer J. & Lago S. (1983) *J. Chem. Phys.* **78**, 5750.
Fischer J. & Lustig R. (1985) *Fluid Phase Equil.* **22**, 245.
Kohler F. (1977) *Ber. Bunsenges. Phys. Chem.* **81**, 1037.
Koide A. & Kihara T. (1974) *Chem. Phys.* **5**, 34.
Lago S., Sevilla P., Padilla P. & Lombardero M. (1985) *Dense Fluids Conf.*, Meeting of the Group of Thermodynamics of The Royal Society of Chemistry, Bristol, England.
Lago S., Sevilla P. & Padilla P. (1986) *2nd Liblice Int. Conf. on Molecular Fluids*, Bechyne, Czechoslovakia.
Lago S. & Padilla P. (1987) To be published.
Lago S. & Sevilla P. (1988) *J. Chem. Phys.* In press.
Lee L. L. & Hulburt H. M. (1973) *J. Chem. Phys.* **44**, 1.
Lipschutz S. (1965) *General Topology*, Chap. 4. Schaum, New York.
Lustig R. (1986) *Mol. Phys.* **59**, 173.
McDonald I. R. (1972) *Mol. Phys.* **23**, 41.
Sevilla P. & Lago S. (1985) *Comput. Chem.* **9**, 39.

Appendix overleaf

APPENDIX

Program Listing

```
      PROGRAM CICLOS
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                    PROGRAM CICLOS                                       *
C THIS PROGRAM HAS BEEN WRITTEN BY CARLOS VEGA AND SANTIAGO LAGO          *
C AND CALCULATES THE SHORTEST DISTANCE BETWEEN LINEAR RODS OF             *
C DIFFERENT LENGTH AS WELL AS SOME ORIENTATIONAL AVERAGES DEFINED          *
C IN  THE TEXT                                                            *
C NUPUNT=NUMBER OF RELATIVE ORIENTATIONS BETWEEN THE TWO SEGMENTS FOR     *
C        EVERY GEOMETRICAL CENTER - GEOMETRICAL CENTER (GC-GC)            *
C        DISTANCE USED FOR CALCULATING THE VALUES OF SOME FUNCTIONS       *
C NUCGCG=NUMBER OF GC-GC DISTANCES WHERE AVERAGE VALUES OF THESE          *
C        FUNCTIONS WILL BE CALCULATED                                     *
C IY=    SEED NUMBER TO GENERATE RANDOM NUMBERS                           *
C PSIG1,PSIG2,PEPS1,PEPS2 = PARAMETERS OF THE KIHARA INTERMOLECULAR       *
C                           POTENCIAL                                     *
C TEMP=  KELVIN TEMPERATURE                                               *
C PL1,PL2= LENGTHS OF THE TWO SEGMENTS                                    *
C R0   =    FIRST VALUE OF THE GC-GC   DISTANCE IN UNITS OF SIGMA         *
C                                                                         *
C DELTAR= FORWARDING STEP FOR  THE VALUE OF THE  GC-GC                    *
C         DISTANCE,IN UNITS OF CORRESPONDING SIGMA PARAMETER (PSIG1,      *
C         PSIG2 OR ITS ARITHMETICAL MEAN )                                *
C                                                                         *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
      IMPLICIT DOUBLE PRECISION (A-Z)
      LOGICAL PROPO1,PROPO2,PROPO3,PROPO4,PROPO5,PROPO6,PROPO7,PROPO8
     1,PROPO9 ,ALGES0,PL1ES0,PL2ES0
      INTEGER NUPUNT,NUCGCG,I,J,K,CONTII,CONTIE,CONTEE,CONTPA,IY
      INTEGER CICLO,PASO,CICORR,IA,IJ,IPUN
      DIMENSION  H(8)      ,PROMID(3,150),PROU(3,150),PROEXP(3,150),
     1           PROU1E(3,150),PROETO (3,150)
C
C NECESSARY INPUT OF DATA
C
      READ(5,*)NUPUNT,NUCGCG,IY
      READ(5,*)PSIG1,PSIG2,PEPS1,PEPS2,TEMP
      READ(5,*)PL1,PL2,R0,DELTAR
      WRITE(1,25)R0,NUCGCG,PSIG1,PSIG2,PEPS1,PEPS2,TEMP,PL1,PL2,NUPUNT
     1     ,IY
   25 FORMAT(1X,'INITIAL VALUE  R0/SIGMA',D10.4,' NUCGCG=  ',I5,
     1 'SIG1=',D10.4/1X,'SIG2=',D10.4,1X,'EPS1=',D10.4,
     1 'EPS2=',D10.4/1X,'TEMP=',D10.4,1X,'L1=',D10.4,
     1 1X,'L2=',D10.4 ,3X,'NUPUNT=',I6,'IY(SEMILLA)=',I6)
C
      PI=DACOS(-1.D0)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                         *
C KBOLT IS THE BOLTZMANN CONSTANT                                         *
C ALGES0 DEFINES A LOGICAL VARIABLE WHICH DISCERNS WHETHER A ROD           *
C HAS LENGTH ZERO (ALGES0 IS THEN TRUE) OR NOT (ALGES0 IS THEN            *
C FALSE)                                                                  *
C                                                                         *
C                                                                         *
C ALGES0 DETERMINES THE VALUE OF PASO                                     *
C                                                                         *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      KBOLT=1.3805D-16
      ALGES0=(PL1.EQ.0..OR.PL2.EQ.0.)
      IF(ALGES0) PASO=2
      IF(.NOT.ALGES0) PASO=1
      PL1ES0=PL1.EQ.0.
      PL2ES0=PL2.EQ.0.
      IF(.NOT.ALGES0) GO TO 34
C *  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THE CASE WHERE ONE OF THE ROD LENGTHS IS EQUAL TO ZERO                   *
C NOW THE PROGRAM STUDIES THE INTERACTION BETWEEN TWO POINTS               *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      IF(PL1ES0) SIGY=PSIG1
      IF(PL2ES0) SIGY=PSIG2
      DO 30 IA=1,NUCGCG
      PROMID(2,IA)=(R0+DELTAR*(IA-1))*SIGY
      MINDIS=PROMID(2,IA)
      IF(PL1ES0) GO TO 27
      CALL UYEXP(TEMP,PSIG2,PSIG2,PEPS2,PEPS2,MINDIS,U,EXP,U1EXP0,EXPTO)
      GO TO 28
```

```
   27   CALL UYEXP(TEMP,PSIG1,PSIG1,PEPS1,PEPS1,MINDIS,U,EXP,U1EXP0,EXPTO)
   28   PROU(2,IA)=U/KBOLT/TEMP
        PROEXP(2,IA)=EXP
        PROU1E(2,IA)=U1EXP0/KBOLT/TEMP
        PROETO(2,IA)=EXPTO/KBOLT/TEMP
   30   CONTINUE
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C                                                                           *
C IF  CICLO=1 THE AVERAGED VALUE OF THE                                     *
C SHORTEST DISTANCE BETWEEN  ROD1-ROD1 AND THE AVERAGED VALUES              *
C OF THE FUNCTIONS DEPENDING ON THE SHORTEST DISTANCE ARE OBTAINED           *
C IF CICLO=2 THE SAME  FOR THE INTERACTION                                  *
C ROD2-ROD2,AND IF CICLO=3 FOR THE INTERACTION  ROD1-ROD2                   *
C                                                                           *
C                                                                           *
C THE RESULTS ARE STORED IN THE ARRAYS:                                     *
C PROMID,PROU,PROEXP,PROU1E,PROETO                                          *
C THE DIMENSIONS OF THESE ARRAYS ARE (3,NUCGCG)                             *
C                                                                           *
C THE AVERAGE VALUE OF THE SHORTEST DISTANCE IS  STORED IN THE FIRST        *
C ARRAY                                                                     *
C                                                                           *
C THE AVERAGE VALUE OF THE REFERENCE   POTENTIAL                            *
C IS STORED IN THE SECOND ARRAY                                             *
C                                                                           *
C THE AVERAGE VALUE OF THE BOLZMANN FACTOR OF THE                           *
C REFERENCE   POTENTIAL IS STORED IN THE THIRD ARRAY                        *
C                                                                           *
C THE AVERAGE VALUE OF THE FUNCTION                                         *
C (U1*EXP(-U0/KT)) IS STORED,WHERE U1 IS THE PERTURBATION POTENTIAL         *
C AND U0 IS THE REFERENCE   POTENTIAL  IN THE FOURTH ARRAY                  *
C                                                                           *
C THE INTERACTION ROD1-ROD1 IS                                              *
C STORED IN THE FIRST ROW OF EVERY ARRAY                                    *
C                                                                           *
C THE INTERACTION ROD2-ROD2 IS                                              *
C STORED IN THE SECOND ROW OF EVERY ARRAY                                   *
C                                                                           *
C THE INTERACTION ROD1-ROD2 IS                                              *
C STORED IN THE THIRD ROW OF EVERY ARRAY                                    *
C                                                                           *
C EVERY COLUMN J STORES  THE AVERAGE VALUE OF THE MAGNITUDE WHICH           *
C GIVES ITS NAME TO THE ARRAY AT THE DISTANCE BETWEEN THE GEOMETRICAL       *
C CENTERS OF THE TWO LINEAR RODS GIVEN BY R=R0+(J-1)*DELTAR                 *
C                                                                           *
C                                                                           *
C                                                                           *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
   34   DO 600 CICLO=1,3,PASO
        IF (.NOT.ALGES0) GO TO 48
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C SEE ABOVE FOR THE MEANING OF ALGES0 AND PASO                              *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
        CICORR=(CICLO+1)/2
        GO TO (35,40) , CICORR
    35  IF(PL2ES0) GO TO 50
        IF(PL1ES0) GO TO 60
    40  IF(PL2ES0) GO TO 70
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THE CASE WHERE THE  LENGTH OF ONE ROD IS EQUAL TO 0                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
        L1=PL2
        SIG1=PSIG2
        EPS1=PEPS2
        L2=PL1
        SIG2=PSIG1
        EPS2=PEPS1
        GO TO 100
   48   GO TO (50,60,70),CICLO
   50   L1=PL1
        SIG1=PSIG1
        EPS1=PEPS1
        L2=PL1
        SIG2=PSIG1
        EPS2=PEPS1
        GO TO 100
```

```
   60   L1=PL2
        SIG1=PSIG2
        EPS1=PEPS2
        L2=PL2
        SIG2=PSIG2
        EPS2=PEPS2
        GO TO 100
   70   L1=PL1
        SIG1=PSIG1
        EPS1=PEPS1
        L2=PL2
        SIG2=PSIG2
        EPS2=PEPS2
  100   G1=L1/2.D0
C
        C1=-(L1/2.D0)
        GAMMA1=L1
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C NOW THE COMPUTATION  OF SHORTEST DISTANCES BEGINS AT EVERY GC-GC      *
C DISTANCE GIVEN                                                        *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
C
        SIGX=(SIG1+SIG2)/2.D0
        R=R0*SIGX
        DO 570 I=1 ,NUCGCG
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C  THE VALUE OF THE NEXT VARIABLES                                       *
C  IS SET EQUAL TO 0 FOR EVERY NEW GC-GC DISTANCE                        *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
        SUEXTO=0.D0
        SUU1E0=0.D0
        SUMMID=0.D0
        SUMU=0.D0
        SUMEXP=0.D0
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THE AVERAGED  VALUES OF THE FUNCTIONS DEPENDING                       *
C ON THE SHORTEST DISTANCE WILL BE CALCULATED FOR A GIVEN               *
C DISTANCE WITH NPUNT ORIENTATIONS                                      *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
        DO 520 J=1,NUPUNT
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C NOW THE PROGRAM GENERATES THE SECOND SEGMENT(ROD).                    *
C THE GEOMETRICAL CENTER IS RANDOMLY GENERATED  AS THE FIRST STEP       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
        FI=2.D0*PI*URAND(IY)
        CGCO =(2.D0*URAND(IY))-1.D0
        CGSE =DSQRT(1.D0  -( DABS( CGCO))**2.D0)
        X=R*CGSE *DCOS(FI)
        Y=R*CGSE *DSIN(FI)
        Z=R*CGCO
        IF(.NOT.(ALGES0.AND.CICLO.EQ.3))GO TO 200
        CALL DISML0(G1,C1,X,Y,Z,MINDIS)
        GO TO 480
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C NOW THE PROGRAM RANDOMLY GENERATES THE ENDS     OF THE SECOND         *
C SEGMENT ,NAMELY ITS RELATIVE ORIENTATION                              *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
  200   FIPR=2.D0*PI*URAND(IY)
        SEGCO =(2.D0*URAND(IY))-1.D0
        SEGSE =DSQRT(1.D0-( DABS(SEGCO))**2.D0)
        XPR=(L2/2.D0)*SEGSE *DCOS(FIPR)
        YPR=(L2/2.D0)*SEGSE *DSIN(FIPR)
        ZPR=(L2/2.D0)*SEGCO
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C (A2,B2,C2) (E2,F2,G2) ARE THE COORDINATES OF THE ENDS OF              *
C THE SECOND SEGMENT                                                    *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
        A2=X+XPR
        B2=Y+YPR
        C2=Z+ZPR
```

```
C
      E2=X-XPR
      F2=Y-YPR
      G2=Z-ZPR
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C ALFA2,BETA2,GAMMA2 ARE THE DIRECTOR  VECTORS OF THE SECOND           *
C SEGMENT                                                              *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      ALFA2= E2-A2
      BETA2=F2-B2
      GAMMA2=G2-C2
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THE PROGRAM WILL CALCULATE THE VALUE (TRUE OR NOT TRUE) OF TWO       *
C LOGIC VARIABLES WHICH WILL BE USEFUL BELOW.                          *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      PROPO1=((-L1/2.D0).LE.C2).AND.(C2.LE.(L1/2.D0))
      PROPO2=((-L1/2.D0).LE.G2).AND.(G2.LE.(L1/2.D0))
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C NOW THE PROGRAM ANALYZE WHETHER THE TWO SEGMENTS ARE PARALLEL  OR    *
C NOT                                                                  *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      AUXI1=DABS(GAMMA2-L2)
      IF(AUXI1.LT.1.D-20) GO TO 210
      GO TO 250
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THE CASE FOR PARALLEL SEGMENTS BEGINS                                *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
  210 P=DMIN1(C2,G2)
      Q=DMAX1(C2,G2)
      PROPO3=(P.LE.(-L1/2.D0)).AND.((-L1/2.D0).LE.Q)
      PROPO4=(P.LE.(L1/2.D0)).AND.((L1/2.D0).LE.Q)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C NOW THE PROGRAM WILL EXAMINATE IF THE TWO RODS HAVE SOME COMMON Z     *
C COORDINATE  RANGE                                                    *
      PROPO5=((PROPO1).OR.(PROPO2)).OR.((PROPO3).OR.(PROPO4))
C THE SUBCASE WHERE Z COORDINATES RANGE OVERLAPS                        *
      IF(PROPO5) MINDIS=DSQRT(A2**2+B2**2)
C THE VALUE OF THE SHORTEST DISTANCE IS ALREADY KNOWN.                 *
C NOW THE PROGRAM IS GOING TO CALCULATE THE VALUES OF THE              *
C FUNCTIONS DEPENDING ON THE SHORTEST DISTANCE                         *
C                                                                      *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IF(PROPO5) GO TO 480
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THE SUBCASE WHERE Z COORDINATES DOES NOT OVERLAP                      *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IF(.NOT.PROPO5) GO TO 280
C
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THE GENERAL CASE FOR NON-PARALLEL SEGMENTS                           *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
  250 MU1=((C2-C1)/GAMMA1)-(1/GAMMA1)*((BETA2*GAMMA2*B2)+(ALFA2
     1     *A2*GAMMA2))*(1/(L2**2-GAMMA2**2))
      MU2=(-1.D0*(A2*ALFA2+B2*BETA2))/(L2**2-GAMMA2**2)
      PROPO6=(0.0D0.LT.MU1).AND.(MU1.LT.1.D0)
      PROPO7=(0.0D0.LT.MU2).AND.(MU2.LT.1.D0)
      PROPO8=(PROPO6).AND.(PROPO7)
      IF(PROPO8) X2=A2-(ALFA2**2*A2+ALFA2*BETA2*B2)/(L2**2-GAMMA2
     1           **2)
      IF(PROPO8)Y2=B2-(ALFA2*BETA2*A2+BETA2**2*B2)/(L2**2-GAMMA2**2)
      IF(PROPO8)MINDIS=DSQRT(X2**2+Y2**2)
      IF(PROPO8)GO TO 480
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THE SHORTEST DISTANCE IS INTERIOR-INTERIOR AND                        *
C THE NEXT SENTENCE IS 480                                             *
C IF ANY OF MU1 OR MU2 IS OUTSIDE (0,1) END-END AND INTERIOR-END     *
C DISTANCES MUST BE COMPUTED                                           *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
C
  280 H(1)=DSQRT(A2**2+B2**2+(C2+(L1/2.D0))**2)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THE 4 DISTANCES END-END ARE CALCULATED.                                    *
C THEY WILL BE CALLED    H(1),H(2),H(3),H(4)                                *
C H(1) HAS BEEN CALCULATED IN THE LAST SENTECE .NOW THE                     *
C OTHER 3 END-END DISTANCES ARE CALCULATED                                  *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      H(2)=DSQRT(A2**2+B2**2+(C2-(L1/2.D0))**2)
      H(3)=DSQRT(E2**2+F2**2+(G2+(L1/2.D0))**2)
      H(4)=DSQRT(E2**2+F2**2+(G2-(L1/2.D0))**2)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C NOW THE DISTANCE FROM AN END POINT  TO THE LINE FORMED                     *
C BY THE OTHER SEGMENT WILL BE CALCULATED. FIRSTLY WE SHALL                  *
C CALCULATE THE DISTANCE FROM  THE END POINTS OF THE                         *
C SECOND SEGMENT (THIS SEGMENT ISN'T AT THE ORIGIN) TO THE                   *
C LINE FORMED BY THE FIRST SEGMENT.                                         *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IF(PROPO1) H(5)=DSQRT(B2**2+A2**2)
      IF(.NOT.PROPO1) H(5)=1.D33
      IF(PROPO2) H(6)=DSQRT(E2**2+F2**2)
      IF(.NOT.PROPO2) H(6)=1.D33
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C NOW WE SHALL CALCULATE THE DISTANCES FROM THE END                          *
C POINTS OF THE FIRST SEGMENT (THIS SEGMENT IS AT THE ORIGIN)                *
C TO THE LINE FORMED BY THE SECOND ONE.                                     *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DO 420 K=1,2

      XINTER=((GAMMA2*A2-C2*ALFA2)*(GAMMA2**2+BETA2**2)-
     1        (BETA2*ALFA2)* (GAMMA2*B2-C2*BETA2)
     1        +((GAMMA2**2)*(ALFA2)*(L1/2.D0)*(-1.D0)**K))
     1        / (GAMMA2**3+ALFA2**2*GAMMA2+BETA2**2*GAMMA2)
C
      MU2INT=(XINTER-A2)/ALFA2
      PROPO9=(0.0D0.LE.MU2INT).AND.(MU2INT.LE.1.D0)
      IF (PROPO9) H(6+K)=(DSQRT((B2    *GAMMA2-BETA2*(C2+(-1.D0)
     1            **(K+1)*(L1/2.D0)))**2+
     1            (A2*GAMMA2-ALFA2*(C2+(-1.D0)**(K+1)*(L1/2.D0)))**2
     1            +(A2*BETA2-ALFA2*B2)**2))
     1            /DSQRT(ALFA2**2+BETA2**2+GAMMA2**2)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C IF THE PARAMETER MU2INT IS OUTSIDE (0,1)                                   *
C AN  ARBITRARILY HUGE VALUE IS GIVEN (1.D33) FOR THE DISTANCE               *
C END-INTERIOR                                                              *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IF (.NOT.PROPO9)  H(6+K)=1.D33
  420 CONTINUE
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THE MINIMUM  OF THE NEXT 8 DISTANCES IS CHOSEN AS                          *
C THE SHORTEST DISTANCE BETWEEN THE TWO SEGMENTS.                            *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      MINDIS=DMIN1(H(1),H(2),H(3),H(4),H(5),H(6),H(7),H(8))
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C NOW THE FUNCTIONS DEPENDING ON THE SHORTEST DISTANCE ARE GOING             *
C TO BE CALCULATED                                                          *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
  480 CALL UYEXP(TEMP,SIG1,SIG2,EPS1,EPS2,MINDIS,U,EXP,U1EXP0,EXPTO)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THE RESULT OF EVERY FUNCTION IS ADDED                                      *
C                                                                           *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      SUMMID=SUMMID+MINDIS
      SUMU=SUMU+U
      SUMEXP=SUMEXP+EXP
      SUU1E0=SUU1E0+U1EXP0
      SUEXTO=SUEXTO+EXPTO
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THE PROGRAM BEGINS THE CALCULATION  OF THE SHORTEST  DISTANCE              *
```

```
C BETWEN TWO SEGMENTS FOR ANOTHER RELATIVE ORIENTATION AT THE          *
C SAME GC-GC DISTANCE                                                  *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
  520 CONTINUE
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THE AVERAGE VALUE IS THE TOTAL SUM  DIVIDED BY THE NUMBER            *
C OF POINTS. POTENTIALS ARE GIVEN IN UNITS OF KBOLT*TEMP               *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      PROMID(CICLO,I)=SUMMID/NUPUNT
      PROU(CICLO,I)=SUMU/NUPUNT/KBOLT/TEMP
      PROEXP(CICLO,I)=SUMEXP/NUPUNT
      PROU1E(CICLO,I)=SUU1E0/NUPUNT/KBOLT/TEMP
      PROETO(CICLO,I)=SUEXTO/NUPUNT/KBOLT/TEMP
C
C
 550  R=R+DELTAR*SIGX
 570  CONTINUE
 600  CONTINUE
      IF(.NOT.(PL1ES0))GO TO 608
      DO 605 IJ=1,NUCGCG
      GUARDA=PROU(1,IJ)
      PROU(1,IJ)=PROU(2,IJ)
      PROU(2,IJ)= GUARDA
C
      GUARDA=PROMID(1,IJ)
      PROMID(1,IJ)=PROMID(2,IJ)
      PROMID(2,IJ)=GUARDA
C
      GUARDA=PROEXP(1,IJ)
      PROEXP(1,IJ)=PROEXP(2,IJ)
      PROEXP(2,IJ)=GUARDA
C
      GUARDA=PROU1E(1,IJ)
      PROU1E(1,IJ)=PROU1E(2,IJ)
      PROU1E(2,IJ)=GUARDA
C
      GUARDA=PROETO(1,IJ)
      PROETO(1,IJ)=PROETO(2,IJ)
      PROETO(2,IJ)=GUARDA
 605  CONTINUE
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C OUTPUT  OF RESULTS                                                   *
C                                                                      *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 608  WRITE(1,620)
 620  FORMAT('R/SIG__PROEXP_____PROU1E_____PROETO_____PROEXP___
     1_____PROU1E_____PROETO_____PROPEXP_____PROU1E_____PROET
     1O_____'   )
      DO 900 J=1,NUCGCG
      R=R0+DELTAR*(J-1.)
      WRITE(1,630)R,PROEXP(1,J),PROU1E(1,J),PROETO(1,J) ,
     1              PROEXP(2,J),PROU1E(2,J),PROETO(2,J) ,
     1              PROEXP(3,J),PROU1E(3,J),PROETO(3,J)
 630  FORMAT(F5.3,1X,9(D13.6,1X))
 900  CONTINUE
      STOP
      END
C
C
      SUBROUTINE DISML0(G1,C1,X,Y,Z,MINDIS)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THIS SUBROUTINE CALCULATES  THE SHORTEST DISTANCE BETWEEN            *
C A SEGMENT (IN THE Z AXIS) AND A POINT                                *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT DOUBLE PRECISION (A-Z)
      DIMENSION DISTAN(2)
      IF(C1.LE.Z.AND.Z.LE.G1) GO TO 111
      DISTAN(1)=DSQRT(X**2+Y**2+(Z-G1)**2)
      DISTAN(2)=DSQRT(X**2+Y**2+(Z-C1)**2)
      MINDIS=DMIN1(DISTAN(1),DISTAN(2))
      GO TO 112
 111  MINDIS=DSQRT(X**2+Y**2)
 112  RETURN
      END
C    ***********************************************
C    ***********************************************
```

```
C
C FUNCTION GENERATING  RANDOM NUMBERS
      DOUBLE PRECISION FUNCTION URAND(IY)
      DOUBLE PRECISION HALFM,DATAN,DSQRT
      DATA M2/0/,ITWO/2/
      IF(M2.NE.0)GO TO 20
      M=1
10    M2=M
      M=ITWO*M2
      IF(M.GT.M2)GO TO 10
      HALFM=M2
      IA=8*IDINT(HALFM*DATAN(1.D0)/8.D0)+5
      IC=2*IDINT(HALFM*(0.5D0-DSQRT(3.D0)/6.D0))+1
      MIC=(M2-IC)+M2
      S=0.5/HALFM
20    IY=IY*IA
      IF(IY.GT.MIC) IY=(IY-M2)-M2
      IY=IY+IC
      IF(IY/2.GT.M2) IY=(IY-M2)-M2
      IF(IY.LT.0) IY=(IY+M2)+M2
      URAND=FLOAT(IY)*S
      RETURN
      END
C
C
      SUBROUTINE UYEXP(TEMP,SIG1,SIG2,EPS1,EPS2,MINDIS,U,EXP,
     1    U1EXP0,EXPTO)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THIS SUBROUTINE COMPUTES THE VALUE OF 4 FUNCTIONS ( U,                    *
C EXP(-U/KT),    U1*EXP(-U /KT),     UTOTAL*EXP(-UTOTAL/KT))               *
C U =REFERENCE    POTENTIAL                                                 *
C U1=PERTURBATION POTENTIAL                                                 *
C UTOTAL=U +U1                                                              *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      IMPLICIT DOUBLE PRECISION (A-Z)
      KBOLT=1.3805D-16
      SIG12=(SIG1+SIG2)/2.D0
      EPS12=DSQRT(EPS1*EPS2)
      SIGMIN=SIG12*((2.D0)**(1.D0/6.D0))
      IF(MINDIS.GE.SIGMIN) U=0.D0
      IF(MINDIS.GE.SIGMIN) EXP=1.D0
      IF(MINDIS.GE.SIGMIN) GO TO 500
      AUXIU=(SIG12/MINDIS)
      IF(AUXIU.GE.1.D4) U=4*EPS12*(1.D48-1.D24)+EPS12
      IF(AUXIU.LE.1.D4)
     1U=EPS12+4*EPS12*((SIG12/MINDIS)**12-(SIG12/MINDIS)**6)
      AUXIE=U/(KBOLT*TEMP)
      IF(AUXIE.GE.140.D0) EXP=0.D0
      IF(AUXIE.LE.140.D0) EXP=DEXP( - AUXIE)
500   IF(MINDIS.LT.SIGMIN) U1=-EPS12
      IF(MINDIS.GE.SIGMIN)
     1    U1=4.D0*EPS12*((SIG12/MINDIS)**12-(SIG12/MINDIS)**6)
      U1EXP0=U1*EXP
510   UTOTAL= U+U1
      AUXIET=UTOTAL/(KBOLT*TEMP)
      IF(AUXIET.GE.140.D0) EXPTO=0.D0
      IF(AUXIET.LT.140.D0) EXPTO=DEXP(-AUXIET)
      EXPTO=EXPTO*UTOTAL
520   RETURN
      END
```